

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-5678

(P2001-5678A)

(43)公開日 平成13年1月12日(2001.1.12)

(51) Int.Cl.⁷
G 0 6 F 9/46
// G 0 6 F 17/50

識別記号
340

F I
G O 6 F 9/46
15/60

テマコート[®](参考)
5B046

340A 5B046
664J 5B098

審査請求 未請求 請求項の数 6 OL (全 12 頁)

(21) 出願番号 特願平11-171886
(22) 出願日 平成11年6月18日(1999.6.18)

(71) 出願人 000005223
富士通株式会社
神奈川県川崎市中原区上小田中4丁目1番
1号

(72) 発明者 安川 武男
神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72) 発明者 福岡 俊之
神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

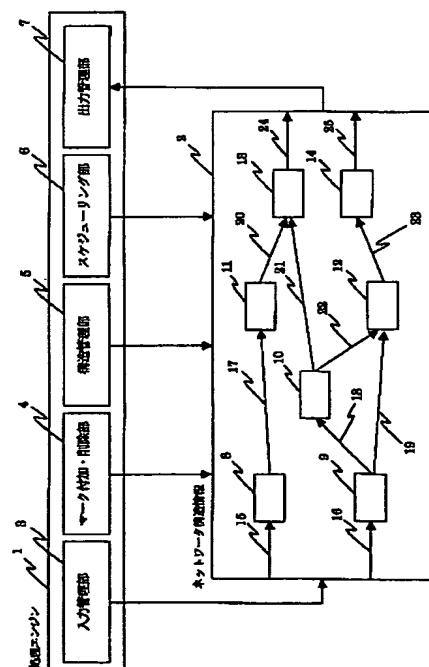
(74) 代理人 100095555
弁理士 池内 寛幸
F ターム(参考) 5B046 DA01 JA04
5B098 GA08 GC01

(54) 【発明の名称】 ネットワーク型情報処理装置及び方法

(57)【要約】

【課題】 ソートを行うことなく、適切な順序で、かつ最小の演算時間で処理することのできるネットワーク型情報処理装置及び方法を提供する。

【解決手段】 外部からの入力をネットワーク構造情報に伝達し、入力が変化したノードに対して出力結果が確定となることを示すマークを付加・削除し、ネットワーク構造情報に含まれるノード及びノード間のリンクを追加・削除し、ノードの再計算のスケジューリングを行い、ネットワーク構造情報の計算結果を出力するネットワーク型情報処理装置であって、入力が変化すると、入力と依存関係を有するノードにマークを付加し、マークが付加されたノードで、入力がマークが付加されていないノードからの出力のみであるノードからスケジューリングして再計算し、再計算が完了したらマークを削除し、マークの付加されたノードがネットワーク構造情報の中に存在しなくなるまで再計算を繰り返す。



1

【特許請求の範囲】

【請求項1】 处理エンジンと、ネットワーク構造情報とで構成されるネットワーク型情報処理装置において、前記処理エンジンが、外部からの入力を前記ネットワーク構造情報に伝達する入力管理部と、前記ネットワーク構造情報に含まれるノードのうち、入力が変化したノードに対して出力結果が不確定となることを示すマークを付加又は削除するマーク付加・削除部と、前記ノードの再計算のスケジューリングを行うスケジューリング部と、前記ネットワーク構造情報の計算結果を出力する出力管理部とで構成され、前記入力管理部からの入力が変化した場合に、前記マーク付加・削除部で前記入力と依存関係を有する前記ノードにマークを付加し、前記マークが付加された前記ノードの中で、入力がマークが付加されていない状態の前記ノードからの出力のみである前記ノードからスケジューリングして再計算し、再計算が完了したらマークが付加されている前記ノードから前記マークを削除し、前記マークの付加された前記ノードが前記ネットワーク構造情報の中に存在しなくなるまで再計算を繰り返すことを特徴とするネットワーク型情報処理装置。

【請求項2】 各前記ノードごとに前回の入力と計算結果を保存し、マークを付加された前記ノードへの入力が前回と一致する場合には再計算を行わずに、前回の計算結果を出力とする請求項1記載のネットワーク型情報処理装置。

【請求項3】 前記ネットワーク構造情報に含まれる二以上の前記ノードを一つのグループとし、前記グループを前記ノードとして取り扱う請求項1又は2記載のネットワーク型情報処理装置。

【請求項4】 前記ネットワーク構造情報に含まれる前記ノード及び前記ノード間のリンクを追加・削除する構造管理部をさらに含み、

前記構造管理部において、各前記ノードの追加・削除を実行時に動的に行う請求項1記載のネットワーク型情報処理装置。

【請求項5】 处理エンジンと、ネットワーク構造情報とを用いるネットワーク型情報処理方法であって、前記処理エンジンにおいて、外部からの入力を前記ネットワーク構造情報に伝達する工程と、

前記ネットワーク構造情報に含まれるノードのうち、入力が変化したノードに対して出力結果が不確定となることを示すマークを付加又は削除する工程と、

前記ノードの再計算のスケジューリングを行う工程と、前記ネットワーク構造情報の計算結果を出力する工程とを含み、

前記入力が変化した場合に、前記入力と依存関係を有する前記ノードにマークを付加し、前記マークが付加され

10

た前記ノードの中で、入力がマークが付加されていない状態の前記ノードからの出力のみである前記ノードからスケジューリングして再計算し、再計算が完了したらマークが付加されている前記ノードから前記マークを削除し、前記マークの付加された前記ノードが前記ネットワーク構造情報の中に存在しなくなるまで再計算を繰り返すことを特徴とするネットワーク型情報処理方法。

【請求項6】 处理エンジンと、ネットワーク構造情報とを用いるネットワーク型情報処理方法を実現するコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体であって、前記処理エンジンにおいて、外部からの入力を前記ネットワーク構造情報に伝達するステップと、前記ネットワーク構造情報に含まれるノードのうち、入力が変化したノードに対して出力結果が不確定となることを示すマークを付加又は削除するステップと、前記ノードの再計算のスケジューリングを行うステップと、

前記ネットワーク構造情報の計算結果を出力するステップとを含み、

前記入力が変化した場合に、前記入力と依存関係を有する前記ノードにマークを付加し、前記マークが付加された前記ノードの中で、入力がマークが付加されていない状態の前記ノードからの出力のみである前記ノードからスケジューリングして再計算し、再計算が完了したらマークが付加されている前記ノードから前記マークを削除し、前記マークの付加された前記ノードが前記ネットワーク構造情報の中に存在しなくなるまで再計算を繰り返すことを特徴とするコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】 【発明の属する技術分野】 本発明は、入出力を有する部品がネットワーク構造に結合し配置されているネットワーク型情報処理装置及び方法に関する。

【0002】 【従来の技術】 従来から、情報処理装置を構築するに当たり、規模が大きく複雑な処理を対象とする場合には、当該処理全体を規模が小さく処理が比較的簡単な複数の単位部品に分割し、それらを連結してネットワーク構造とすることで当該情報処理装置を構築する方法が良く用いられている。かかる方法は、問題解決の手段として、あるいは処理の単位部品の再利用を計るという観点からも注目されている方法の一つである。

【0003】 このようなネットワーク型情報処理装置においては、常にデータの流れが存在するため、単純に単位部品を連結しただけでは、正常に動作しない。したがって、データの流れに適合させて、適切な順序で単位部品ごとの処理を行う必要がある。

【0004】 従来は、かかる処理順序の制御に、単位部

20

品間の依存関係に着目してソートを行う方法が良く用いられている。例えば、ネットワークの依存関係から単位部品についてトポロジカルソートを行った後、ソート後の順序で演算していくという方法である。

【0005】

【発明が解決しようとする課題】しかし、上述したようなソートを用いる方法では、単位部品の追加や結合関係の変化等のネットワーク構造自体が変化する場合には、再度全体をソートし直す必要が生じる。特にネットワーク構造が巨大な場合には、かかる再ソート作業に要する時間は無視できるレベルの時間ではない。したがって、実際に、演算実行時に動的にネットワーク構造を変化させるということは困難であるという問題点を内包していた。

【0006】本発明は、上記問題点を解決するために、ソートを行うという独立したフェーズを有することなく、適切な順序で、かつ最小の演算時間で処理することのできるネットワーク型情報処理装置及び方法を提供することを目的とする。

【0007】

【課題を解決するための手段】上記目的を達成するために本発明にかかるネットワーク型情報処理装置は、処理エンジンと、ネットワーク構造情報とで構成されるネットワーク型情報処理装置であって、処理エンジンが、外部からの入力をネットワーク構造情報に伝達する入力管理部と、ネットワーク構造情報に含まれるノードのうち、入力が変化したノードに対して出力結果が不確定となることを示すマークを付加又は削除するマーク付加・削除部と、ノードの再計算のスケジューリングを行うスケジューリング部と、ネットワーク構造情報の計算結果を出力する出力管理部とで構成され、入力管理部からの入力が変化した場合に、マーク付加・削除部で入力と依存関係を有するノードにマークを付加し、マークが付加されたノードの中で、入力がマークが付加されていない状態のノードからの出力のみであるノードからスケジューリングして再計算し、再計算が完了したらマークが付加されているノードからマークを削除し、マークが付加されたノードがネットワーク構造情報の中に存在しなくなるまで再計算を繰り返すことを特徴とする。

【0008】かかる構成により、マークの付加及び削除によって再計算の必要であるノードであるか否かを制御でき、再計算の必要であるノードのみ再計算すれば適切な順序で、しかも最小の演算時間で処理を実行することが可能となる。

【0009】また、本発明にかかるネットワーク型情報処理装置は、各ノードごとに前回の入力と計算結果を保存し、マークを付加されたノードへの入力が前回と一致する場合には再計算を行わずに、前回の計算結果を出力することが好ましい。ノードへの入力が前回の入力と一致していれば、当該ノードからの出力についても変化

しないので、再度演算処理を行うことを回避することで、計算機資源の無駄遣いを未然に防止することができ、ひいては演算処理時間の短縮化を図ることができるからである。

【0010】また、本発明にかかるネットワーク型情報処理装置は、ネットワーク構造情報に含まれる二以上のノードを一つのグループとし、グループをノードとして取り扱うことが好ましい。複数のノードを一つのノードと認識することで、考慮すべきリンクの数を減少させることができ、演算処理時間の短縮化を図ることができるからである。また、ノードの組合せによって、新たな機能を有するノードを生成することが容易となり、プログラムの部品化を促進することもできるからである。

【0011】また、本発明にかかるネットワーク型情報処理装置は、ネットワーク構造情報に含まれるノード及びノード間のリンクを追加・削除する構造管理部をさらに含み、構造管理部において、各ノードの追加・削除を実行時に動的に行なうことが好ましい。連続して演算結果を出力するアプリケーションにおいては、出力が停止したり、非連続になることは好ましくないからである。

【0012】また、本発明は、上記のようなネットワーク型情報処理装置の機能をコンピュータの処理ステップとして実行するソフトウェアを特徴とするものであり、具体的には、処理エンジンにおいて、外部からの入力を前記ネットワーク構造情報に伝達する工程と、ネットワーク構造情報に含まれるノードのうち、入力が変化したノードに対して出力結果が不確定となることを示すマークを付加又は削除する工程と、ノードの再計算のスケジューリングを行う工程と、ネットワーク構造情報の計算結果を出力する工程とを含み、入力が変化した場合に、入力と依存関係を有するノードにマークを付加し、マークが付加されたノードの中で、入力がマークが付加されていない状態のノードからの出力のみであるノードからスケジューリングして再計算し、再計算が完了したらマークが付加されているノードからマークを削除し、マークの付加されたノードがネットワーク構造情報の中に存在しなくなるまで再計算を繰り返すネットワーク型情報処理方法並びにそのような工程をプログラムとして記録したコンピュータ読み取り可能な記録媒体であることを特徴とする。

【0013】かかる構成により、コンピュータ上へ当該プログラムをロードさせ実行することで、マークの付加及び削除によって再計算の必要であるノードであるか否かを制御でき、再計算の必要であるノードのみ再計算すれば適切な順序で、しかも最小の演算時間で処理を実行することが可能となるネットワーク型情報処理装置を実現することが可能となる。

【0014】

【発明の実施の形態】(実施の形態1)以下、本発明の実施の形態1にかかるネットワーク型情報処理装置につ

いて、図面を参照しながら説明する。図1は本発明の実施の形態1にかかるネットワーク型情報処理装置のブロック構成図である。図1において、1は処理エンジンを、2はネットワーク構造情報を示す。

【0015】処理エンジン1は、入力された情報をネットワーク構造情報へ伝える入力管理部3と、ネットワーク構造情報2に含まれるノードにマークを付加又は削除するマーク付加・削除部4と、ノードやリンクの生成・削除等を行う構造管理部5と、ノード再計算の順序を正しくスケジューリングするスケジューリング部6と、ネットワーク構造情報2からの出力を取得して出力する出力管理部7で構成されている。

【0016】また、ネットワーク構造情報2は、任意の数の入出力を有するノードと、それらの依存関係を示す方向性を有するリンクとで構成されている。例えば、リンク17はノード8からの出力に依存しており、ノード8の出力が変化した場合には、ノード11を再計算する必要が生じること、及びノード8がノード11よりも先に出力を決定しないなければならないことを意味している。

【0017】まず、入力管理部3において、入力された情報をネットワーク構造情報2に伝える。ここで、入力が一定不变の場合には、ネットワーク構造情報2の出力も一定となる。例えば、画面に表示されているキャラクタの行動決定に本実施の形態1にかかる装置を用いる場合には、外部の環境が変わらない限りキャラクタの選択する行動は変化しない。したがって、入力管理部3では、ネットワーク構造情報2にかかる入力を伝えるか否かについても制御することができる。こうすることで、不要な演算を回避することができ、計算機資源の無駄遣いを防止することができる。

【0018】次に、入力の変化を検知した場合においては、マーク付加・削除部4が起動される。マーク付加・削除部4は、ノードの出力が未決定であるノード、すなわち変化した入力によって再計算を行なう必要が生じる可能性のあるノードについてマークを付与する。図2に、入力16が変化した場合のネットワーク構造情報のマーク状況図を示す。なお、マーク付加・削除部4は、変化があった入力から依存するノードを追跡して依存関係のあるノードにマークを付与していくことになるが、この場合の付与効率を考えると、以下に示すような深さ優先探索に従ってマークを付与していくことが好ましい。

【0019】例えば図2において、入力16はノード9にリンクしているので、まずノード9にマークが付与される。次に、ノード9はリンク18を経てノード10と依存関係があるので、ノード10にもマークが付与される。

【0020】ここで、ノード9はリンク19を経てノード12とも依存関係を有するが、同時にノード12はノード10ともリンク22を通じて依存関係を有してい

る。本発明においては、ネットワークにおけるリンク階層の深さが浅いノードを優先して探索を行いながらマークを付与していくので、ノード12はノード10よりもリンク階層が深いノードであると判断され、ノード12についてはこの時点でのマーク付与対象とはならない。

【0021】そして、ノード10にマークが付与されたことで、リンク21を経てノード13にマークが付与される。ノード13の出力24はネットワーク構造情報2の出力であるので、これ以上深く探索することができない。そこで、1リンクだけ前に戻って、ノード10からリンク22を経てノード12にマークが付与される。

【0022】以下、同様にしてノード14についてもマークを付与することで、図2におけるハッチング部にマークを付与することになる。なお、ノード12はノード9からリンク19を経ることによってもマーク付与され得るが、この時点で既にマークが付与された状態となっているので、特にこれより先のマーク状態を考慮する必要はない。

【0023】次に、構造管理部5では、ネットワーク構造情報2内におけるノードの新たな追加や削除等を行う。ノードを追加して新たなリンクを構築する際に、既存のノード及びリンクに対しては何ら影響は与えないようにしておき、新たな判断基準の追加、いわゆる学習条件の追加に該当するようなノードの追加を動的に行なうことができる。すなわち、ノード追加時にプログラム自身を停止する必要もなく、実行中であっても新たなノードを追加することができる。

【0024】さらに、スケジューリング部6において、マークの付加されているノードのうち、入力がマークが付与されていない状態のノードからの出力のみであるノードを探索してスケジューリングする。例えば図2においては、かかる状態のノードはノード9に限定される。

【0025】スケジューリングされたノードから再計算を行うことになる。ここで、スケジューリングされたノードへの入力が、前回の入力と比較して変化していない場合と変化している場合について以下のように処理を行う。

【0026】まず、入力が変化していない場合には、前回の計算結果を再利用することで、全体の演算量を減ずることが可能となり、ひいては計算機資源の無駄遣いの防止につながる。

【0027】一方、入力が変化している場合には、ノードを再評価することで、新しい出力を計算する。再計算すると同時に、マーク付加・削除部4によって再計算を行なったノードのマークが解除される。図2においては、ノード9の再計算が終わったらノード9のマークを解除することになる。

【0028】すべてのノードについて、マークが解除されているのを確認できるまで、再計算を繰り返す。そして、マークが完全に解除された時点の演算結果を出力管

理部で出力する。

【0029】図3から図5に上述したマークに関する処理の進行状況を図示する。すなわち、まず図3では、ノード9がスケジュールされてマークを解除した状態を示している。ここで、次にスケジュール可能なノードを探索すると、ノード10とノード12が考えられるが、どちらを先にスケジュールしても演算の最終結果には影響しない。

【0030】そこで、まずノード10をスケジュールした状態を図4に示す。図4において、次にスケジュール可能なノードは、ノード12とノード13であるので、これらのノードについても並列に、あるいは順にスケジュールした状態を図5に示す。

【0031】図5において、最後に残っているノードはノード14だけとなっているので、最後にノード14をスケジュールすることで、すべてのノードについて、マークが解除されていることになる。

【0032】なお、本実施の形態1において、ノードへの入力が変化したか否かの判断には、例えばノードのスケジュール時に前回の演算結果と比較する方法を用いることができる。ただし、かかる方法に限定されるものではなく、同様の効果を有する方法で有ればどのような方法でも良い。例えば、ノードの演算結果出力が変化した場合に参照している他のノードにフラグを立てるといった方法であっても良い。例えば、図2において、ノード9を再評価した結果、演算出力値が変化した場合には、ノード10とノード12に入力が変化していることを示すフラグを設定するといった方法も考えられる。

【0033】また、入力が変化しない場合においては、ノード出力も変化しないことから再計算は不要となるが、入力が変化した場合であってもノード出力が変化しない場合も考え得る。例えば、図6に示すような単純なノードを考えると、このノードへの入力値がたとえ1から100に変化したとしても出力値は常に“TRUE”である。したがって、入力値が変化したとしてもその出力値が変化する場合もあれば変化しない場合もあるので、出力値のみを判断基準として、再計算の必要性を判断すれば良い。

【0034】次に、本実施の形態1にかかるネットワーク型情報処理装置の一実施例について図面を参考しながら説明する。図7は、本発明の一実施例にかかるネットワーク型情報処理装置の構成図である。図7は、本実施の形態1にかかるネットワーク型情報処理装置を仮想生物の感情・行動選択装置へ応用したものである。図7において、27は画像表示装置を、28は処理エンジンを、29はネットワーク構造情報を、30は入力管理装置を、31はマーク付加・削除装置を、32は構造管理装置を、33はスケジューラを、34は出力管理装置を、それぞれ示す。

【0035】まず、画像表示装置27は仮想生物によっ

て選択された行動を実際に画像で表現する装置を示し、コンピュータディスプレイやLCD等が良く用いられる。画像表示装置27は、仮想生物が選択した行動を表示すると同時に、表示の結果についても計算し、例えば仮想生物の位置や仮想生物が存在する世界の情報である明るさや気温、風向き等についても出力することが可能となっている。

【0036】また、処理エンジン28は、画像表示装置27からの外部の状態を外部情報ノード35を介することでネットワーク構造情報29へ伝える。そして、ネットワーク構造情報29において、内部の状態を計算している。例えば外部の明るさノード36の出力が変化したり、あるいは仮想生物の活動力ノード37の出力が変化することによって、仮想生物の感情・行動が選択される。

【0037】本実施例では、例えば眠気ノード38は、外部が暗くて仮想生物の活動力が低い場合に大きくなるという行動決定モデルを示している。同様に、空腹感ノード39は、活動力が高い状態が継続されると上昇していくという感情決定モデルも実現している。行動選択ノード40は、これらの情報を総合的に判断して優先順位をつけて、次に行うべき行動を出力する。

【0038】また、図8は、図7に示すネットワーク型情報処理装置におけるネットワーク構造情報29に構造管理装置32が新たなノードとして幸福感ノード41を追加したものである。ここでは、眠気ノード38と空腹感ノード39からの入力と行動選択ノード40への出力が新たに設定されており、既存のノードについての依存関係については何ら影響を与えていない。従って、「幸福感」という行動選択上の新たな判断基準が追加された形になっていることがわかる。

【0039】さらに、図9においては、目的地選択ノード44を追加して行動選択ノード40とリンクしたものである。この場合、入力された距離が一定の距離よりも近くなると、当該位置を目的地とする。この目的地選択ノード44に対する入力として、例えば餌までの距離ノード42や、巣までの距離ノード43等に、構造管理装置32が切り替えることで、状況に応じた目的地の変更を行うことが可能となる。

【0040】もちろん、構造管理装置32の切り替えによっては、目的地選択ノード44における判断基準のみが変化するのであるから、既存のノードについての依存関係には何ら影響を与えていない。したがって、構造管理装置32の切り替えは、プログラムの実行中であっても、プログラムを停止することなく動的に行うことが可能である。

【0041】以上のように本実施の形態1によれば、ネットワーク構造情報の更新を、ノードのついてソートするという独立したプロセスを経ることなく行うことができ、ノード間の依存関係を自由に変更することが可能と

なる。また、ノードの追加・削除等によってネットワーク構造に変化が生じた場合であっても、処理を中断することなく連続して実行することが可能となる。

【0042】(実施の形態2) 次に本発明の実施の形態2について、図面を参照しながら説明する。

【0043】図10は本発明の実施の形態2にかかるネットワーク型情報処理装置におけるネットワーク構造情報2の例示図である。図10においては、ネットワーク構造情報2の一部をグループ化し、そのグループ26を一つのノードとして取り扱っている。

【0044】具体的には、ノード10、ノード11、ノード12を一つのグループ26とし、グループ26を入力が17及び18、出力が20、23である一つのノードとして取り扱う。この場合、例えば入力16が変化した場合には、グループ化しない状態においてはノード9、ノード10、ノード12、ノード13、ノード14の5つのノードにマークが付加されていた。

【0045】これを、グループ化して図11に示すように一つのノードとして内部を隠蔽すると、同じく入力16が変化した場合には、ノード9、ノード26、ノード13、ノード14の4つのノードにマークが付加されることになり、ノード構造が単純化することでマーク数が減少する。

【0046】また、ノード9の出力に変化が無かった場合には、ノード26の内部についていっさいスケジューリングする必要が生じない。したがって、グループ化することによって、全体の演算量を減少させることができるという利益を享受することができる。

【0047】以上のように本実施の形態2によれば、複数のノードを一つのノードと認識することで、考慮すべきリンクの数を減少させることができ、演算処理時間の短縮化を図ることができる。また、ノードの組合せによって、新たな機能を有するノードを生成することができるとなり、プログラムの部品化を促進することも期待できる。

【0048】次に、本発明の実施の形態1にかかるネットワーク型情報処理装置を実現するプログラムの処理の流れについて説明する。図12に本発明の実施の形態1にかかるネットワーク型情報処理装置を実現するプログラムの処理の流れ図を示す。

【0049】まず、入力管理部において、新たな入力を検出すると(ステップS121)、当該入力に応じて、入力されるノードと依存関係にあるノードについてマーク付加・削除部がマークを付与する(ステップS122)。

【0050】次に、スケジューリング部がマークの付与されたノードの中から入力がマークの付与されていないノードからの出力のみであるノードを探査し、探索されたノードをスケジューリングする(ステップS123)。かかるノードの処理を最初に行う必要があるから

である。入力直後においては、入力されたノードがスケジューリングされることになる。

【0051】そして、スケジューリングされたノードへの入力の内容について、当該ノードへの前回に行われた入力と変化があるか否かについて判断する(ステップS124)。前回の入力内容と変化がない場合には、今回の出力についても前回と同様の出力値となるので(ステップS124: No)、前回の演算結果を出力値として利用することで再計算を行わない(ステップS125)。こうすることで、部だな演算時間を省略して演算時間の短縮化を図ることができるからである。一方、前回の入力内容と変化がある場合には(ステップS124: Yes)、かかる入力の内容に基づいて再計算を行い、新たな出力値を算出することになる(ステップS126)。

【0052】入出力の行われたノードについては、マーク付加・削除部によって付与されたマークが解除される(ステップS127)。当該ノードについての演算処理が完了したことを示すためである。そして、ネットワーク構造情報内の全てのノードについてマークが解除されるまで上記処理を繰り返し(ステップS128: No)、全てのノードについてマークが解除された場合には(ステップS128: Yes)、出力管理部において演算結果を出力することになる(ステップS129)。

【0053】本発明の実施の形態にかかるネットワーク型情報処理装置を実現するプログラムを記憶した記録媒体は、図13に示す記録媒体の例に示すように、CD-ROM132-1やフロッピーディスク132-2等の可搬型記録媒体132だけでなく、通信回線の先に備えられた他の記憶装置132や、コンピュータ133のハードディスクやRAM等の記録媒体134のいずれでも良く、プログラム実行時には、プログラムはローディングされ、主メモリ上で実行される。

【0054】また、本発明の実施の形態にかかるネットワーク型情報処理装置により生成された各ノードにおける前回の演算処理結果等を記録した記録媒体も、図9に示す記録媒体の例に示すように、CD-ROM132-1やフロッピーディスク132-2等の可搬型記録媒体132だけでなく、通信回線の先に備えられた他の記憶装置131や、コンピュータ133のハードディスクやRAM等の記録媒体134のいずれでも良く、例えば本発明にかかるネットワーク型情報処理装置を利用する際にコンピュータ133により読み取られる。

【0055】

【発明の効果】以上のように本発明にかかるネットワーク型情報処理装置によれば、ネットワーク構造情報の更新を、ノードのついてソートするという独立したプロセスを経ることなく行うことができ、ノード間の依存関係を自由に変更することができる。また、ノードの追加・削除等によってネットワーク構造に変化が生じた場

合であっても、処理を中断することなく連続して実行することが可能となる。することができる。

【0056】さらに、本発明にかかるネットワーク型情報処理装置によれば、マークの付加及び削除によって再計算の必要であるノードであるか否かを制御でき、再計算の必要であるノードのみ再計算すれば適切な順序で、しかも最小の演算時間で処理を実行することが可能となる。

【0057】また、本発明にかかるネットワーク型情報処理装置によれば、複数のノードを一つのノードと認識することで、考慮すべきリンクの数を減少させることができ、演算処理時間の短縮化を図ることができる。さらに、ノードの組合せによって、新たな機能を有するノードを生成することが容易となり、プログラムの部品化を促進することも期待できる。

【図面の簡単な説明】

【図1】 本発明の実施の形態1にかかるネットワーク型情報処理装置の構成図

【図2】 本発明の実施の形態1にかかるネットワーク型情報処理装置におけるネットワーク構造情報の説明図

【図3】 本発明の実施の形態1にかかるネットワーク型情報処理装置におけるマーク処理の例示図

【図4】 本発明の実施の形態1にかかるネットワーク型情報処理装置におけるマーク処理の例示図

【図5】 本発明の実施の形態1にかかるネットワーク型情報処理装置におけるマーク処理の例示図

【図6】 本発明の実施の形態1にかかるネットワーク型情報処理装置におけるノードの例示図

【図7】 本発明の一実施例にかかるネットワーク型情報処理装置の構成図

【図8】 本発明の他の実施例にかかるネットワーク型情報処理装置の構成図

【図9】 本発明の他の実施例にかかるネットワーク型情報処理装置の構成図

【図10】 本発明の実施の形態2にかかるネットワーク型情報処理装置におけるネットワーク構造情報の例示図

*図

【図11】 本発明の実施の形態2にかかるネットワーク型情報処理装置におけるネットワーク構造情報の例示図

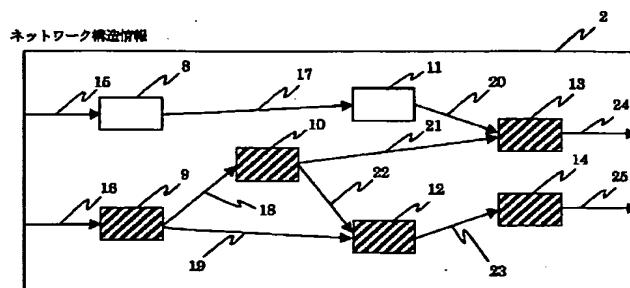
【図12】 本発明の実施の形態1にかかるネットワーク型情報処理装置における処理の流れ図

【図13】 記録媒体の例示図

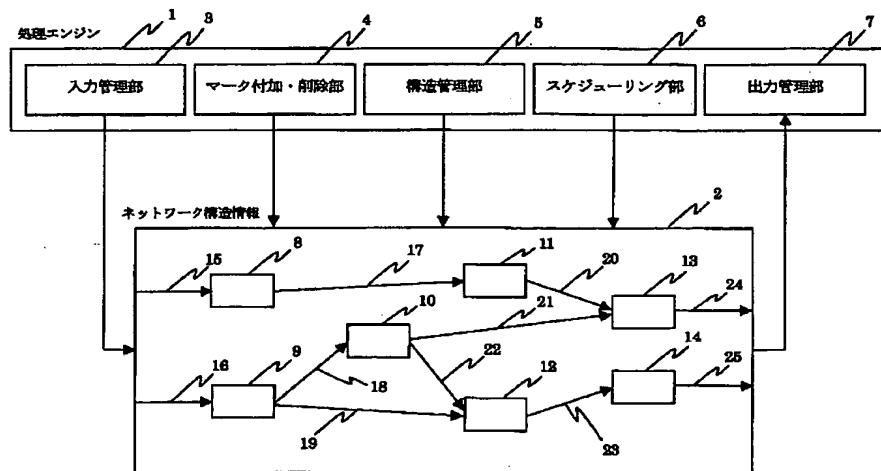
【符号の説明】

- | | |
|----|--|
| 10 | 1、28 処理エンジン |
| | 2、29 ネットワーク構造情報 |
| | 3 入力管理部 |
| | 4 マーク付加・削除部 |
| | 5 構造管理部 |
| | 6 スケジューリング部 |
| | 7 出力管理部 |
| 20 | 8、9、10、11、12、13、14、35、36、
37、38、39、40、41、42、43、44 ノード |
| | 15、16 入力 |
| | 17、18、19、20、21、22、23 リンク |
| | 24、25 出力 |
| | 26 グループ |
| | 27 画像表示装置 |
| | 30 入力管理装置 |
| | 31 マーク付加・削除装置 |
| | 32 構造管理装置 |
| | 33 スケジューラ |
| | 34 出力管理装置 |
| 30 | 131 回線先の記憶装置 |
| | 132 CD-ROMやフロッピーディスク等の可搬型記録媒体 |
| | 132-1 CD-ROM |
| | 132-2 フロッピーディスク |
| | 133 コンピュータ |
| | 134 コンピュータ上のRAM/ハードディスク等の記録媒体 |

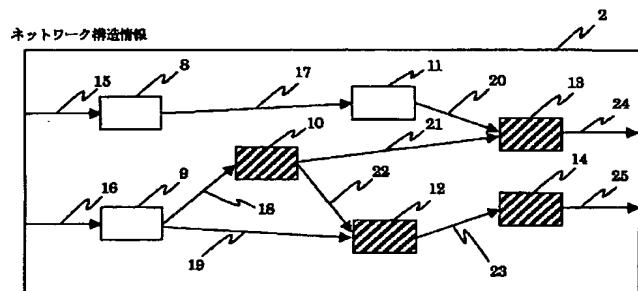
【図2】



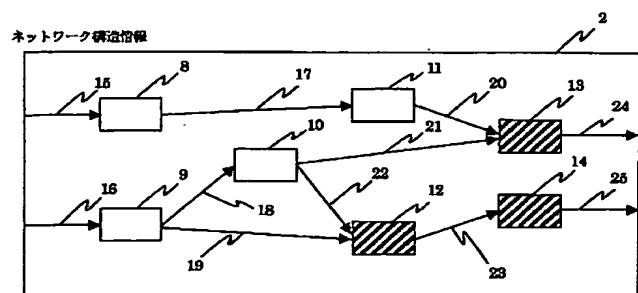
【図1】



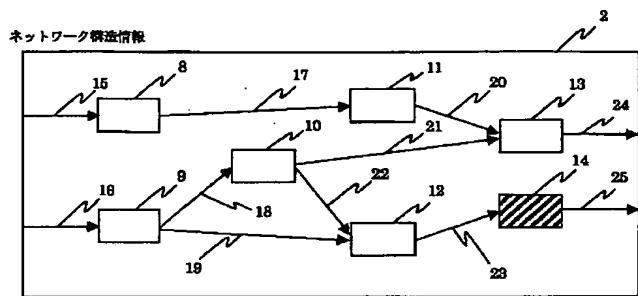
【図3】



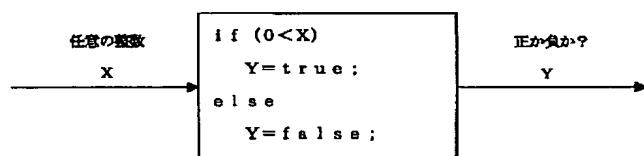
【図4】



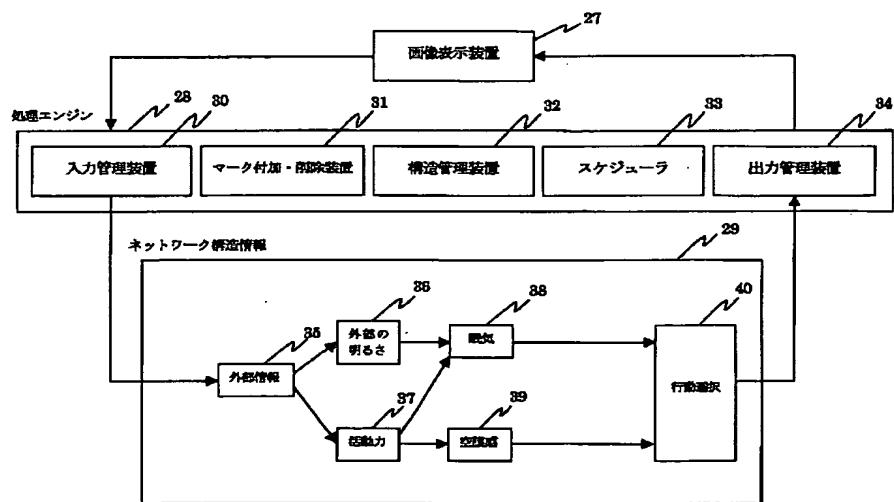
【図5】



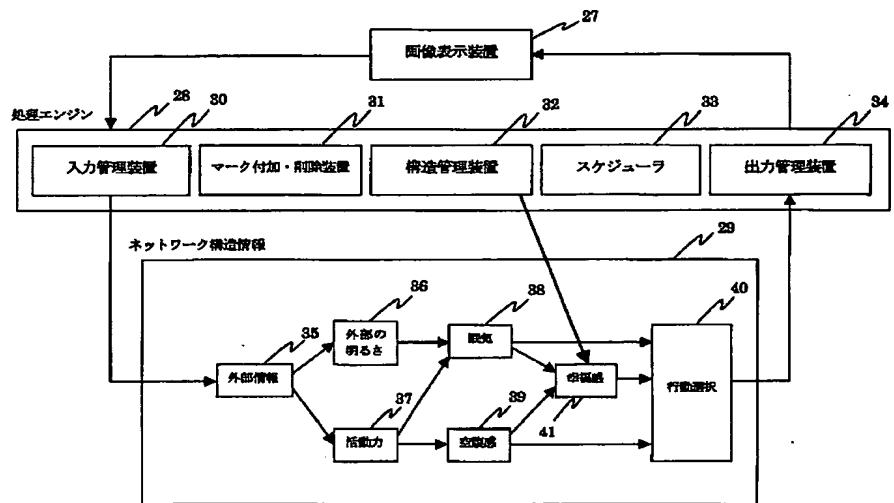
【図6】



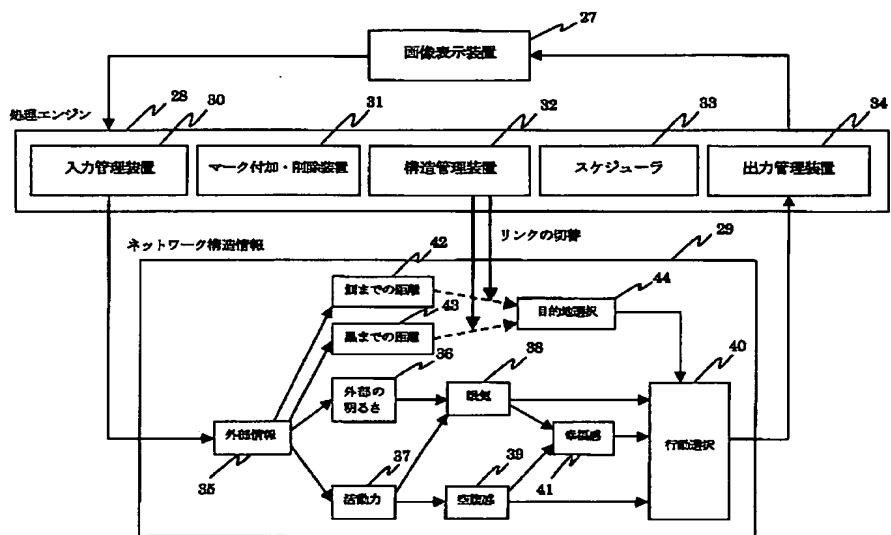
【図7】



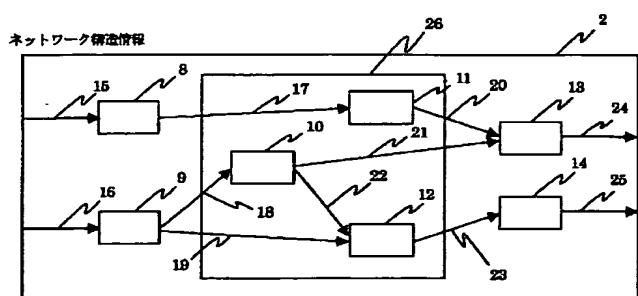
【図8】



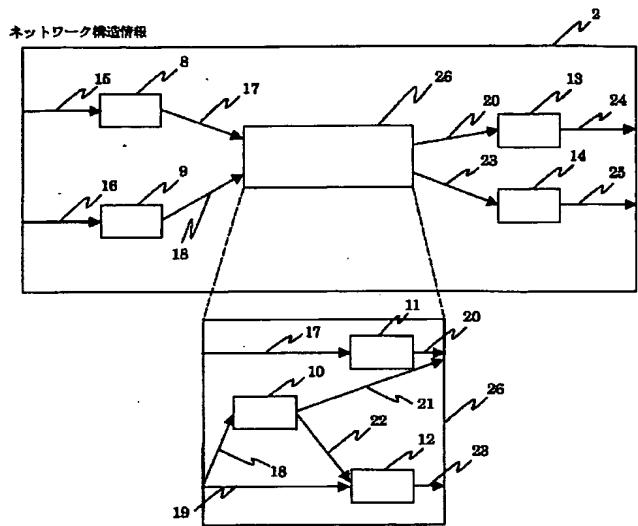
【図9】



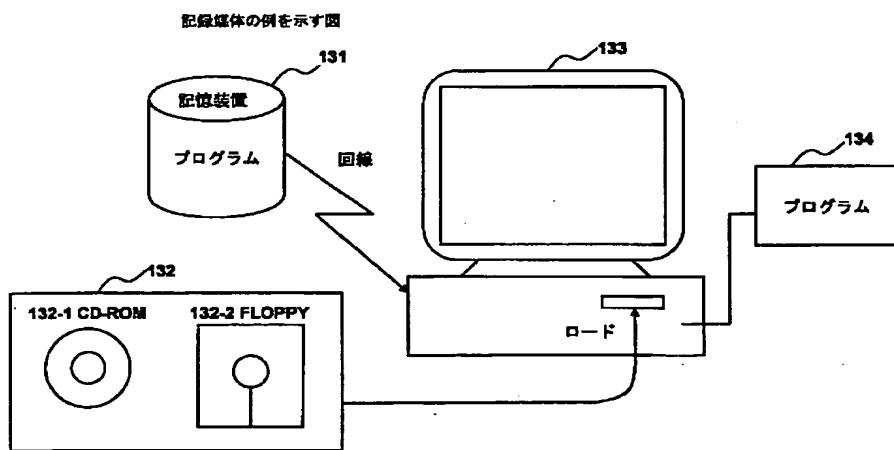
【図10】



【図11】



【図13】



【図12】

